#19 4-21-03

Patent
237/117
21039-7096

CERTIFICATE OF MAILING (37 C.F.R. § 1.8(a))

I hereby certify that this paper (along with any referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date indicated below with sufficient postage as First Class Mail in an envelope addressed to: Box Appeal Brief, Commissioner for Patents, Washington, D.C. 20231:

Dated: April 3, 2003

Linda Major
Linda Major

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Wei Ming Hu et al.

Serial No.: 09/223,660

Filed: December 30, 1998

For: METHOD AND SYSTEM FOR DIAGNOSTIC PRESERVATION OF THE STATE OF A COMPUTER SYSTEM

Group Art Unit: 2189

Examiner: Tim T. Vo

APPEAL BRIEF UNDER 37 CFR § 1.192

Box APPEAL BRIEF
Commissioner for Patents
Washington, D.C. 20231

The applicants submit this Appeal Brief pursuant to the Notice of Appeal filed in this case on January 31, 2003. This brief is submitted in triplicate.

I. REAL PARTY IN INTEREST

The real party in interest is the assignee Oracle International Corporation of Redwood Shores, California.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-72 are pending in the subject application. Appeal is taken from the Examiner's rejection of claims 1-72.

IV. STATUS OF AMENDMENTS AFTER FINAL REJECTION

A final Office action was mailed on October 1, 2002. In response to the final Office action, applicants filed an amendment after final rejection on December 2, 2002. None of the claims were amended in the December 2, 2002 amendment. The Examiner indicated in an Advisory Action mailed on December 30, 2002 that the request for reconsideration has been considered but does not place the application in condition for allowance.

V. SUMMARY OF THE INVENTION

The present invention is defined by the pending claims and their equivalents. The present section of the Appeal Brief is set forth merely to comply with the requirements of 37 C.F.R. § 1.192(c)(5) and is not intended to limit the pending claims in any way. See M.P.E.P. § 1206.

An embodiment of a method of diagnosing a computer system after a failure is illustrated in Figure 2. Rather than dumping the system memory and restoring the system to its original state when system failure occurs, the state of a first set of system resources are preserved in place after failure occurs in the computer system (208). (Specification pg. 8, ll. 11-14; pg. 9, ll. 7-15). The computer system is accessed by utilizing a second set of system resources (210). (Specification pg. 8, ll. 14-17; pg. 9, ll. 17-19). Failure is then diagnosed by analyzing one or more resources from the first set of system resources (210). (Specification pg. 9, ll. 16-17).

VI. ISSUE

The issue presented is whether claims 1-72 are patentable over U.S. Patent No. 5,485,573 to Tandon in view of U.S. Patent No. 6,195,760 to Chung et al. and further in view of U.S. Patent No. 4,164,017 to Randell et al. under 35 U.S.C. § 103(a).

VII. GROUPING OF THE CLAIMS

Claims 1-72 stand or fall together. Claim 1 is the representative claim.

VIII. ARGUMENTS

In the October 1, 2002 Office action, the Examiner rejected claims 1-72 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,485,573 to Tandon (hereinafter "Tandon") in view of U.S. Patent No. 6,195,760 to Chung et al. (hereinafter "Chung") and further in view of U.S. Patent No. 4,164,017 to Randell et al. (hereinafter "Randell").

Claim 1 is representative of the rejected claims. Claim 1 recites "preserving in place the state of a first set of system resources after the failure occurs in the computer system." (emphasis added). The Examiner states in the October 1, 2002 Office action:

> From the examiner's view and interpretations that a computer system comprises a storage for storing original information state. When there is a failure occurs in a computer, the information is restored from the storage the saved the original information. Randell teaches this concept. In the background, Randell teaches, a memory for storing all items information in the system so these items information restored as desired after a failure occurs (see column 1 lines 8-22). This teaching anticipates the claim limitation. Further, Randell found that this method is extremely

> wasteful of storage capacity. Therefore, Randell improvement is to minimize the storage capacity and still retrieving original state information efficiently by determining whether there is a need to preserve the state of each item of information (see column 1 lines 44-50) (emphasis added).

(October 1, 2002 final Office action, pgs. 2-3, para. 3). As specifically pointed out by the Examiner, Randell teaches storing information on the original state of a system and then restoring the system to an original state using the stored information when failure occurs. In contrast, claim 1 recites "preserving in place the state of a first set of system resources after the failure occurs in the computer system." (emphasis added). Accordingly, Randell teaches away from the claim element.

Nevertheless, the Examiner maintains that Randell discloses the claim element. Specifically, the Examiner states in the December 30, 2002 Advisory Action:

> Randell teaches memory for storing all items information in the system so these information restored as desired after a failure occurs (column 1 lines 47-50, column 11 lines 41-50, column 17 lines 49-50 and figure 10 cache store 36 and register E). This teaching teaches "preserve in place the state of the items of information after the failure occurs in the computer system.

(December 30, 2002 Advisory Action, pg. 2). However, Randell discloses:

> A data processor has provision for error recovery by the division of a program into a set of program blocks. The apparatus has a main store and a cache store that preserves the states that have previously been taken up by items of information prior to commencing each block so that the states of the items of information prevailing at the beginning of a block can be restored if required.

(Abstract). Hence, the object of the invention in Randell is error recovery, i.e., to restore systems to their original states when failure occurs by storing the initial states of the systems; it is not to preserve "in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1. Further, Randell discloses:

> Whether an acceptance test is passed or failed, local declarations made during the block which caused success or failure must be discarded to put the system in the same condition as it was in when the block started . . . .

(Col. 3, ll. 8-11). Again, Randell teaches away from "preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1 since it discloses discarding local declarations, regardless of whether there is a failure or not, to put the system in the same condition as it was when the block started. Additionally, Randell discloses:

> [O]nly the values of the variables on entry to a recovery block need to be preserved, there [is] no need to preserve any intermediate values of variables which change more than once during a recovery block . . . .

(Col. 3, ll. 43-47). Once more, Randell emphasizes that intermediate values of variables are not preserved. Whereas, claim 1 recites "preserving in place the state of a first set of system resources after the failure occurs in the computer system." Randell also discloses:

> States 51 to 57 . . . are used to restore values of variables which had been modified in the recovery block after failure of an acceptance test. . . . the cache values recorded in the top region of the cache store 36 are re-written into the main store so as to erase the effects of the erroneous block.

(Col. 11, ll. 41-48). Yet again, Randell teaches erasing the effects of an erroneous block and restoring the system back to its original state. Randell does not disclose or suggest "preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1. Moreover, Randell discloses:

> States 51 to 59 perform the reverse operation and are used to restore values of variables which have been modified in the recovery block after failure of an acceptance test.

(Col. 17, ll. 46-49). Time and again, Randell teaches restoring systems to their original states and discarding and/or erasing intermediate states of the systems. Thus, not only does Randell fail to teach or suggest "preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1, Randell teaches away from the claim element.

Tandon is directed to "the detection and identification of errors in programs which execute on multiple host processors and more particularly to the collection of the necessary data for determining the source of errors in a multi-host data base management system." (Col. 1, ll. 21-25). Tandon does not disclose "preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1.

Chung is directed to the "detection of a failure of an application module running on a host computer on a network and recovery from that failure." (Col. 1, ll. 20-22). "Preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1 is not disclosed anywhere in Chung.

Therefore, even if Randell, Tandon, and Chung were combined, the combination neither teaches nor discloses "preserving in place the state of a first set of system resources after the failure occurs in the computer system" as recited in claim 1. Accordingly, applicants respectfully submit that claim 1 is patentable over Tandon in view of Chung and further in view of Randell.

IX. CONCLUSION

For the above reasons, applicants respectfully submit that rejection of claims 1-72 based on 35 U.S.C. § 103(a) has been overcome. Accordingly, applicants request that the Board of Patent Appeals and Interferences overrule the Examiner and allow claims 1-72.

Respectfully submitted,

Bingham McCutchen LLP

Dated: 4-3-03

By:
Erin C. Ming
Reg. No. 47,797

Three Embarcadero Center, Suite 1800
San Francisco, CA 94111-4067
Telephone: (650) 849-4904
Telefax: (650) 849-4800

APPENDIX A: Claims Appealed

1. A method of diagnosing a computer system after a failure comprising:

preserving in place the state of a first set of system resources after the failure occurs in the computer system;

accessing the computer system by utilizing a second set of system resource; and

diagnosing the failure by analyzing one or more resources from the first set of system resources.

2. The method of claim 1 further comprising:

maintaining one or more lists of the first set of system resources.

3. The method of claim 2 in which the one or more lists are linked lists.

4. The method of claim 1 in which the first set of system resources comprise processing entities.

5. The method of claim 4 in which the processing entities comprise processes which are categorized into process types.

6. The method of claim 5 in which the act of preserving the state of the first set of the system resources comprises suspending the state of one or more of the processes in the first set of system resources.

7. The method of claim 6 in which the one or more processes to suspend are suspended by being entered into an idle loop.

8. The method of claim 6 in which the one or more processes to suspend are suspended by an operating system scheduler.

9. The method of claim 6 in which the one or more processes to suspend are selected based upon their process type.

10. The method of claim 1 in which the second set of system resources comprise system resources that have been set aside for diagnostic purposes.

11. The method of claim 1 in which the second set of system resources comprises redundant hardware/software components.

12. A method of diagnosing a computer system after a failure comprising:

detecting a failure on a first computer system;

implementing fail-over to a second computer system after detecting the failure on the first computer system;

preserving in place the state of one or more resources on the first computer system;

accessing the first computer system to diagnose the failure; and

diagnosing the failure by analyzing the one or more resources.

13. The method of claim 12 further comprising:

maintaining a redundant system component for the first computer system; and

shifting control of the redundant system component to the second computer system after the failure.

14. The method of claim 13 in which the redundant system component comprises a disk drive.

15. The method of claim 12 in which the one or more resources comprise one or more processing entities.

16. The method of claim 15 in which the one or more processing entities comprise processes.

17. The method of claim 15 in which the act of preserving the state of one or more resources on the first computer system comprises suspending the one or more processing entities.

18. The method of claim 17 in which the one or more processing entities are suspended by being entered into an idle loop.

19. The method of claim 17 in which the one or more processing entities to suspend are suspended by an operating system scheduler.

20. The method of claim 12 further comprising the act of categorizing the failure into a failure type, and in which the failure type corresponds to the choice of the one or more resources to suspend.

21. A medium readable by a processor, the medium being stored thereon a sequence of instructions which, when executed by the processor, causes the execution of a process of preserving the state of a computer system after a failure by performing:

preserving in place the state of a first set of system resources after the failure occurs in the computer system;

accessing the computer system by utilizing a second set of system resource; and

diagnosing the failure by analyzing one or more resources from the first set of system resources.

22. A medium readable by a processor, the medium being stored thereon a sequence of instructions which, when executed by the processor, causes the execution of a process of preserving the state of a computer system after a failure by performing:

detecting a failure on a first computer system;

implementing fail-over to a second computer system after detecting the failure on the first computer system;

preserving in place the state of one or more resources on the first computer system;

accessing the first computer system to diagnose the failure; and

diagnosing the failure by analyzing the one or more resources.

23. The medium of claim 21, wherein the process further comprises:

maintaining one or more lists of the first set of system resources.

24. The medium of claim 23 in which the one or more lists are linked lists.

25. The medium of claim 21 in which the first set of system resources comprise processing entities.

26. The medium of claim 25 in which the processing entities comprise processes which are categorized into process types.

27. The medium of claim 26 in which the act of preserving the state of the first set of the system resources comprises suspending the state of one or more of the processes in the first set of system resources.

28. The medium of claim 26 in which the one or more processes to suspend are suspended by being entered into an idle loop.

29. The medium of claim 26 in which the one or more processes to suspend are suspended by an operating system scheduler.

30. The medium of claim 26 in which the one or more processes to suspend are selected based upon their process type.

31. The medium of claim 21 in which the second set of system resources comprise system resources that have been set aside for diagnostic purposes.

32. The medium of claim 21 in which the second set of system resources comprises redundant hardware/software components.

33. The medium of claim 22, wherein the process further comprises:

maintaining a redundant system component for the first computer system; and

shifting control of the redundant system component to the second computer system after the failure.

34. The medium of claim 33 in which the redundant system component comprises a disk drive.

35. The medium of claim 22 in which the one or more resources comprise one or more processing entities.

36. The medium of claim 35 in which the one or more processing entities comprise processes.

37. The medium of claim 35 in which the act of preserving the state of one or more resources on the first computer system comprises suspending the one or more processing entities.

38. The medium of claim 37 in which the one or more processing entities are suspended by being entered into an idle loop.

39. The medium of claim 37 in which the one or more processing entities to suspend are suspended by an operating system scheduler.

40. The medium of claim 22 further comprising the act of categorizing the failure into a failure type, and in which the failure type corresponds to the choice of the one or more resources to suspend.

41. A system for diagnosing a computer system after a failure comprising:

a preservation module for preserving in place the state of a first set of system resources after the failure occurs in the computer system;

an access module for accessing the computer system by utilizing a second set of system resources; and

a diagnosis module for diagnosing the failure by analyzing one or more resources from the first set of system resources.

42. The system of claim 41 further comprising:

a list maintenance module, for maintaining one or more lists of the first set of system resources.

43. The system of claim 42 in which the one or more lists are linked lists.

44. The system of claim 41 in which the first set of system resources comprise processing entities.

45. The system of claim 44 in which the processing entities comprise processes which are categorized into process types.

46. The system of claim 45 in which the preservation module preserves the state of the first set of the system resources by suspending the state of one or more of the processes in the first set of system resources.

47. The system of claim 46 in which the one or more processes to suspend are suspended by being entered into an idle loop.

48. The system of claim 46 in which the one or more processes to suspend are suspended by an operating system scheduler.

49. The system of claim 46 in which the one or more processes to suspend are selected based upon their process type.

50. The system of claim 41 in which the second set of system resources comprise system resources that have been set aside for diagnostic purposes.

51. The system of claim 41 in which the second set of system resources comprises redundant hardware/software components.

52. A system for diagnosing a computer system after a failure comprising:

a failure detection module, for detecting a failure on a first computer system;

a fail-over module, for implementing fail-over to a second computer system after detecting the failure on the first computer system;

a resource preservation module, for preserving in place the state of one or more resources on the first computer system;

an access module for accessing the first computer system to diagnose the failure; and

a diagnosis module, for diagnosing the failure by analyzing the one or more resources.

53. The system of claim 52 further comprising:

a redundant system component for the first computer system; and

a control module for shifting control of the redundant system component to the second computer system after the failure.

54. The system of claim 53 in which the redundant system component comprises a disk drive.

55. The system of claim 52 in which the one or more resources comprise one or more processing entities.

56. The system of claim 55 in which the one or more processing entities comprise processes.

57. The system of claim 55 in which the preservation module preserves the state of one or more resources on the first computer system by suspending the one or more processing entities.

58. The system of claim 57 in which the one or more processing entities are suspended by being entered into an idle loop.

59. The system of claim 57 in which the one or more processing entities to suspend are suspended by an operating system scheduler.

60. The system of claim 52 further comprising a categorization module for categorizing the failure into a failure type, in which the failure type corresponds to the choice of the one or more resources to suspend.

61. The method of claim 1, wherein preserving in place the state of a first set of system resources comprises freezing the state of the first set of system resources.

62. The method of claim 1, wherein preserving in place the state of a first set of system resources does not require any copying of the state of the first set of system resources.

63. The method of claim 12, wherein preserving in place the state of one or more resources comprises freezing the state of the first set of system resources.

64. The method of claim 12, wherein preserving in place the state of one or more resources does not require any copying of the state of the one or more resources.

65. The medium of claim 21, wherein preserving in place the state of a first set of system resources comprises freezing the state of the one or more resources.

66. The medium of claim 21, wherein preserving in place the state of a first set of system resources does not require any copying of the state of the first set of system resources.

67. The medium of claim 22, wherein preserving in place the state of one or more resources comprises freezing the state of the one or more resources.

68. The medium of claim 22, wherein preserving in place the state of one or more resources does not require any copying of the state of the one or more resources.

69. The system of claim 41, wherein preserving in place the state of a first set of system resources comprises freezing the state of the one or more resources.

70. The system of claim 41, wherein preserving in place the state of a first set of system resources does not require any copying of the state of the first set of system resources.

71. The system of claim 52, wherein preserving in place the state of one or more resources comprises freezing the state of the one or more resources.

72. The system of claim 52, wherein preserving in place the state of one or more resources does not require any copying of the state of the one or more resources.